

DataFlow Analysis Assignment

Amanjot Singh matr. 152792

Leonardo Temperanza matr. 152831

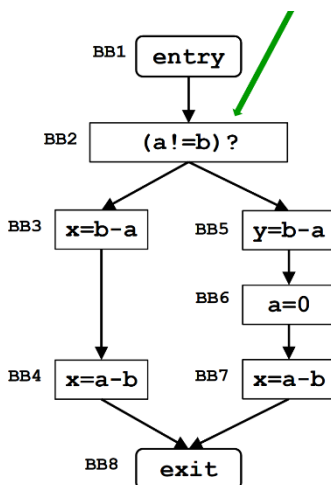
1. Very Busy Expressions

Domain	Set of expressions
Direction	Backward $In[b] = f_b(out[b])$ $Out[b] = \bigwedge in[succ(b)]$
Transfer function	$f_b(x) = Gen_b \cup (x - Kill_b)$
Meet Operator (\wedge)	\cap
Boundary Condition	$In[exit] = \emptyset$
Initial interior points	$In[b] = U$

1. Una espressione $x \oplus y$ é **very busy** se in un punto p se, indipendentemente dal percorso preso da p, l'espressione viene usata prima che uno dei suoi operandi venga definito.
2. Un blocco **genera** l'espressione $x \oplus y$ se valuta $x \oplus y$ e non ridefinisce prima x o y
3. Un blocco **uccide** l'espressione $x \oplus y$ se ridefinisce prima x o y

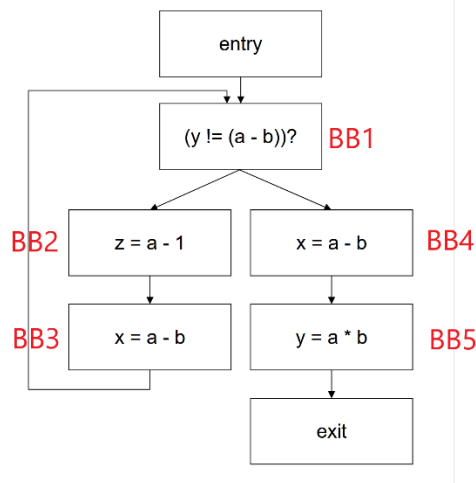
Bit vector composto da 3 bit che indicano le seguenti espressioni:

1. $a - b$
2. $b - a$
3. $a \neq b$



	Iterazione 1		Iterazione 2	
	OUT[B]	IN[B]	OUT[B]	IN[B]
BB7	<000>	<100>	<000>	<100>
BB6	<100>	<000>	<100>	<000>
BB5	<000>	<010>	<000>	<010>
BB4	<000>	<100>	<000>	<100>
BB3	<100>	<110>	<100>	<110>
BB2	<010>	<011>	<010>	<011>
BB1	<011>		<011>	

Se si ragiona limitatamente all'esempio precedente, si potrebbe pensare di inizializzare gli **initial interior points** a \emptyset . Ciò è dovuto alla mancata presenza di loop nel CFG analizzato. In seguito si riporta un controesempio che dimostra che gli **initial interior points** vanno inizializzati ad **U** invece di \emptyset .



Il bit vector è composto da 3 bit che indicano le seguenti espressioni:

1. $a - b$
2. $a - 1$
3. $a * b$

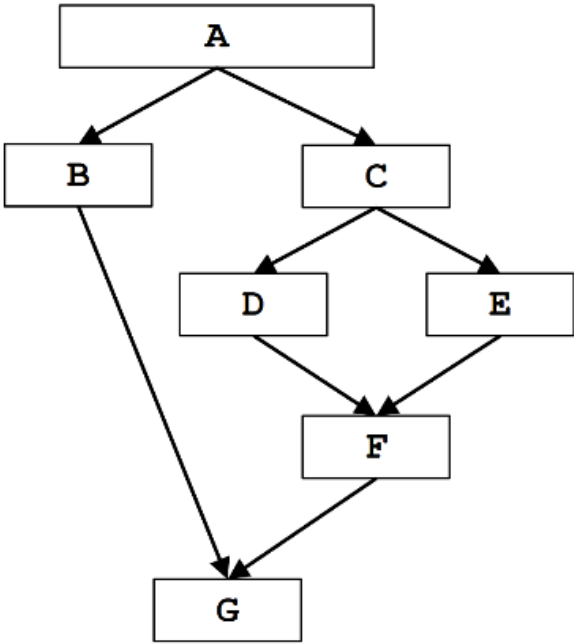
	Iterazione 1		Iterazione 2		Iterazione 3	
	OUT[B]	IN[B]	OUT[B]	IN[B]	OUT[B]	IN[B]
BB5	<0 0 0>	<0 0 1>	<0 0 0>	<0 0 1>	<0 0 0>	<0 0 1>
BB4	<0 0 1>	<1 0 1>	<0 0 1>	<1 0 1>	<0 0 1>	<1 0 1>
BB1	<0 0 0>	<1 0 0>	<1 0 0>	<1 0 0>	<1 0 0>	<1 0 0>
BB3	<1 0 0>	<1 0 0>	<1 0 0>	<1 0 0>	<1 0 0>	<1 0 0>
BB2	<1 0 0>	<1 1 0>	<1 0 0>	<1 1 0>	<1 0 0>	<1 1 0>
ENTRY	<1 0 0>		<1 0 0>		<1 0 0>	

Come si può vedere dalla tabella precedente, viene persa l'informazione sul fatto che l'espressione $a * b$ è **very busy**.

2. Dominator Analysis

Domain	Set of basic blocks
Direction	Forward $\text{Out}[b] = f_b(\text{in}[b])$ $\text{In}[b] = \Lambda \text{ out}[\text{pred}(b)]$
Transfer function	$f_b(x) = b \cup x$
Meet Operator (Λ)	\cap
Boundary Condition	$\text{Out}[\text{entry}] = \emptyset$
Initial interior points	$\text{Out}[b] = U$

1. Un nodo X domina un altro nodo Y se il nodo X appare in ogni percorso del grafo che porta dal blocco ENTRY al blocco Y
2. Per definizione un nodo domina sé stesso



	Iterazione 1		Iterazione 2	
	IN[B]	OUT[B]	IN[B]	OUT[B]
A	<0000000>	<1000000>	<0000000>	<1000000>
B	<1000000>	<1100000>	<1000000>	<1100000>
C	<1000000>	<1010000>	<1000000>	<1010000>
D	<1010000>	<1011000>	<1010000>	<1011000>
E	<1010000>	<1010100>	<1010000>	<1010100>
F	<1010000>	<1010010>	<1010000>	<1010010>
G	<1000000>	<1000001>	<1000000>	<1000001>
EXIT	<1000001>		<1000001>	

3. Constant Propagation

Domain	Set of couple <variable, constant>
Direction	Forward $\text{Out}[b] = f_b(\text{In}[b])$ $\text{In}[b] = \bigwedge \text{out}[\text{pred}(b)]$
Transfer function	$f_b(x) = \text{Gen}_b \cup (x - \text{Kill}_b)$
Meet Operator (\bigwedge)	\cap
Boundary Condition	$\text{Out}[\text{entry}] = \emptyset$
Initial interior points	$\text{Out}[b] = U$

1. Se abbiamo la coppia $\langle x, c \rangle$ al nodo n , significa che x è garantito avere il valore c ogni volta che n viene raggiunto durante l'esecuzione del programma.
2. Un nodo n genera la coppia $\langle x, c \rangle$, se ad x :
 - a. Viene assegnato un valore costante
 - b. Viene assegnato il risultato di un operazione, i cui operandi sono costanti. Un operando è costante se è una costante numerica oppure se è garantito avere un valore c .
3. Un nodo n uccide una coppia $\langle x, c \rangle$ se ridefinisce x .

